What does LIME really see in images?

Damien Garreau[1], Dina Mardaoui[2]

[1]Université Côte d'Azur, LJAD, Inria
[2]Polytech Nice

October 7, 2022

Outline

1. Introduction
2. A primer on LIME for images
3. Theoretical analysis
4. Conclusion

1. Introduction

Setting

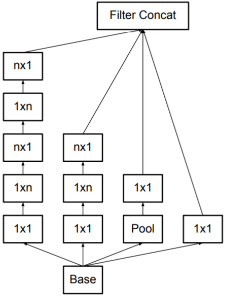
- **Goal:** from input $x \in \mathcal{X}$, predict $y \in \mathcal{Y}$ as $f(x)$
- **Running example:** image classification:

$x =$  $\longmapsto$ $y = \text{trailer\_truck}$

- $f : \mathcal{X} \to \mathcal{Y}$ is a function, potentially *very* complicated
- f_θ corresponds to some architecture choice, $\theta \in \Theta$ parameters
- **Idea:** good choice $\theta^\star$ learned from data

Example: Inception network

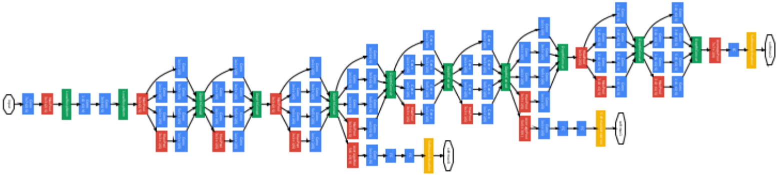
- **Example:** the InceptionV3 network for image classification[1]
- here is one block of the architecture:



[1] Szegedy et al., *Rethinking the inception architecture for computer vision*, CVPR, 2016

Example: Inception network, ctd.

- then stack all these modules (159 layers, 24M parameters)



The need for interpretability

- **Fact 1:** state-of-the-art = deep neural networks
- often referred to as "black-boxes":
  - even more parameters than previous example (*e.g.*, GPT-3, 175 billions[2])
  - even more complicated architectures
- **Fact 2:** machine learning algorithms are now used for *critical* decisions:
  - credit scoring
  - college admissions
  - ...
- **Problem:** we have no idea how a specific decision was made
- **Example 1:** our model achieves good results in the lab, but fails in production (*e.g.*, learning artifacts in the image)
- **Example 2:** social acceptability ("why was I denied a loan?", possible coming legal requirement[3])

[2] Brown et al., *Language models are few-shot learners*, arxiv, 2020

[3] Wachter, Mittelstadt, Floridi, *Why a right to explanation of automated decision-making does not exist in the general data protection regulation*, International Data Privacy Law, 2017

This talk

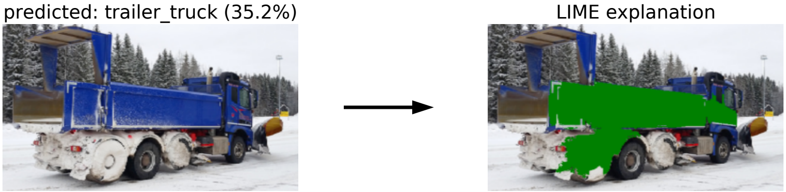
- **This talk** = post hoc, local interpretability
- some recent results about **L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations (LIME[4])
- **Question:** can we analyze it and prove / disprove that it makes sense in simple cases?
- in truth, several versions of the method, depending on the nature of the data:
  - *images* ($\leftarrow$ this talk)
  - *text data*
  - *tabular data*
- complicated operating procedure, but very popular at the moment
- other main contender is SHAP[5]
- image data: $\xi \in \mathbb{R}^{H\times W\times 3}$ an image to explain and $f : \mathbb{R}^{H\times W\times 3} \to [0, 1]$ the model
- **Example:** f is the prediction for a certain class given by InceptionV3

[4] Ribeiro et al., *"Why should I trust you?" Explaining the Prediction of any Classifier*, SIGKDD, 2016
[5] Lundberg and Lee, *A unified approach to interpreting model predictions*, NeurIPS, 2017

2. A primer on LIME for images

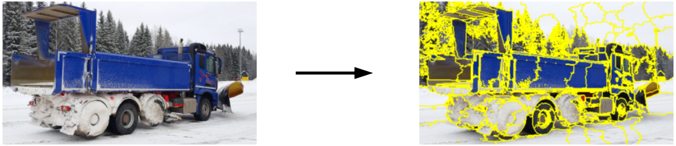
Image LIME

- on a high level, Image LIME operates as follows:
  1. decompose ξ in d *superpixels* (small, homogeneous patches);
  2. create a number of *perturbed samples* (= new images) $x_1, \ldots, x_n$;
  3. *weight* the perturbed samples;
  4. *query* the model, getting predictions $y_i = f(x_i)$;
  5. build a *local surrogate model* $\hat{\beta}_n$ fitting the y_is on the presence or absence of superpixels.
- generally, highlight in the original image the (top 5) positive superpixels:



Step 1: superpixels

- **Interpretable features:** superpixels $J_1, \ldots, J_d$ = sets of pixel indices
- $\cup_k J_k = [H] \times [W]$ and for all $j \neq k$, $J_k \cap J_\ell = \emptyset$
- by default, *quickshift*[6] is used



- in this example, $D = 299 \times 299 \times 3 = 268{,}203$ and $d = 65$ superpixels

[6] Vedaldi and Soatto, *Quickshift and kernel methods for mode seeking*, ECCV, 2008

Step 2: sampling

- **Simple idea:** take a replacement color (say black), randomly switch on and off the superpixels:

- by default, $n = 5000$
- **Potential problem:** black can be a meaningful color for the model (we are trying to *remove* a feature)

Step 2: sampling, ctd.

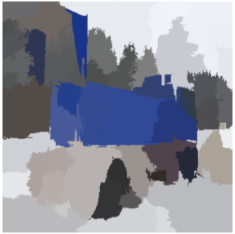
- **Idea:** compute the *mean* of the image on each superpixel
- formally,

$$\forall u \in J_k, \qquad \overline{\xi}_u = \frac{1}{|J_k|} \sum_{u \in J_k} \xi_u \,.$$

- channel-wise if RGB image

segmented ξ

$\overline{\xi}$



Step 2: sampling, ctd.

- **Same idea:** replace superpixels randomly by corresponding superpixel of $\overline{\xi}$

x_1 x_2 … x_n

- **Intuition:** replace the superpixel with something non-informative, but not too far away from the local pixel distribution
- this is the *default* behavior

Step 3: weights

- to each perturbed sample x_i corresponds a binary vector $z_i \in \{0,1\}^d$
- $z_{i,j} = 1$ iff superpixel j is "switched on"
- $\mathbf{1}$ corresponds to ξ
- formally, for all $1 \leq i \leq n$, x_i receives the weight

$$\pi_i := \exp\left(\frac{-\delta(z_i, \mathbf{1})^2}{2\nu^2}\right) ,$$

where δ is the *cosine distance* and $\nu > 0$ is a *bandwidth parameter* (default value $= 0.25$)

Definition: for any two vectors $u, v \in \mathbb{R}^d$, define the *cosine distance* between u and v by

$$\delta(u, v) := 1 - \frac{u^\top v}{\|u\| \cdot \|v\|} .$$

Step 3: weights, ctd.

- **Idea:** give more weight to samples near ξ

original image ξ

x_1

x_2

- many superpixels "switched off" in x_2
- $\Rightarrow$ far away from the original image
- $\Rightarrow$ small weight

Step 4: query

- compute $y_i = f(x_i)$ for every $i \in \{1, \ldots, n\}$

$y_1 = f(x_1) = 0.01$

$y_2 = f(x_2) = 0.04$

...

$y_n = f(x_n) = 0.55$

- cost $= \mathcal{O}(n)$ (n calls to the model)
- generally the main computational cost
- **Remark:** can be parallelized

Step 5: local surrogate model

- finally, train a *local surrogate model*
- by default, *weighted ridge regression*[7]:

$$\hat{\beta}_n \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^{d+1}} \left\{ \sum_{i=1}^{n} \pi_i (y_i - \beta^\top z_i)^2 + \lambda \left\| \beta \right\|^2 \right\} ,$$

  with $\lambda > 0$ a *regularization constant*
- each superpixel receives a coefficient β_j
- **Intuition:** if $\beta_j \gg 0$, superpixel j has a positive influence on the prediction
- **Computational cost:** n queries, then ridge for $n \times d$ data with $n \gg d$: $\mathcal{O}\left(d^2 n\right)$
- **Remark:** a lot of flexibility in the LIME framework, another model / penalty could be used

[7] Hoerl and Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics, 1970

3. Theoretical analysis

Image LIME theory

- **Main question underlying this work:**

LIME operating procedure is complicated, does it make sense for simple models?

- complicated question, some simplifications:
  - $\lambda = 0$ (no penalty)
  - f is bounded
- **Why is this justified?**
- default implementation of ridge is used, $\lambda = 1$
- typically, $n = 5000$ and $d = 50$, thus $\lambda \|\beta\|^2$ is small with respect to the empirical risk
- bounded model is always satisfied by restricting the input space

A first result

- we can show that the explanations stabilize around a limit value when n is large

Proposition (G. and Mardaoui, 2021):[8] Assume that $\lambda = 0$ and that f is bounded. Then, as the number of perturbed samples n goes to infinity, $\hat{\beta}_n \xrightarrow{\mathbb{P}} \beta$, where $\beta \in \mathbb{R}^{d+1}$ is a vector depending only on f, ξ, and ν.

- *Idea of the proof:* $\hat{\beta}_n$ solution of a weighted least square problem, exploitable closed-form + concentration inequalities. $\square$
- **Consequence:** we can focus on β to get insights on LIME
- **Good news:** the expression of β is explicit!

[8] Garreau and Mardaoui, *What does LIME really see in images?*, ICML, 2021

Expression of β

- **Recall:** $z_{i,j} = 1$ if superpixel j is "switched on" in example i
- **Notation:** in the following, z random variable such that the z_is are i.i.d. z, associated x, π

Proposition (G. and Mardaoui, 2021): There exist constants c_d, σ_1, σ_2, and σ_3 such that,

$$\forall 1 \leq j \leq d, \quad \beta_j^f = c_d^{-1} \left\{ \sigma_1 \mathbb{E}\left[\pi f(x)\right] + \sigma_2 \mathbb{E}\left[\pi z_j f(x)\right] + \sigma_3 \sum_{\substack{k=1 \\ k \neq j}}^{d} \mathbb{E}\left[\pi z_k f(x)\right] \right\} .$$

- c_d, σ_1, σ_2, and σ_3 can be computed in closed-form and do not depend on f
- *Proof:* see next slides.

Computing the limit explanation

- **Idea:** with $\lambda = 0$, weighted least squares
- explanations given by
$$\hat{\beta}_n = (Z^\top W Z)^{-1} Z^\top W y \,,$$
with $Z_{i,j} = z_{i,j}$ and $W_{i,i} = \pi_i$
- when n is large,
$$\frac{1}{n} Z^\top W Z \approx \mathbb{E}\left[Z^\top W Z\right] =: \Sigma \quad \text{and} \quad \frac{1}{n} Z^\top W y \approx \mathbb{E}\left[Z^\top W y\right] =: \Gamma \,.$$
- **Key computation:**
$$\Sigma_{j,k} = \mathbb{E}\left[\sum_{i=1}^{n} \pi_i z_{i,j} z_{i,k}\right] \,.$$
- **Key quantity:**
$$\boxed{\alpha_p := \mathbb{E}\left[\pi z_1 \cdots z_p\right] \,.}$$

Computation of the α coefficients

- we can compute the α coefficients in closed-form:

Proposition (G. and Mardaoui, 2021): Let $d \geq 2$ and $p \geq 0$. For any $\nu > 0$, it holds that

$$\alpha_p = \frac{1}{2^d} \sum_{s=0}^{d} \binom{d-p}{s} \cdot \exp\left(\frac{-(1-\sqrt{1-s/d})^2}{2\nu^2} \right) .$$

- *Proof:* conditioning with respect to the number of deletions then combinatorics. $\square$
- large bandwidth:

$$\alpha_p \approx \frac{1}{2^{p-1}} .$$

Σ matrix

- **Recall:** $\alpha_p := \mathbb{E}\left[\pi z_1 \cdots z_p\right]$
- with this notation:

$$\Sigma = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_1 & \cdots & \alpha_1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_3 \\ \alpha_1 & \alpha_3 & \alpha_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \alpha_3 \\ \alpha_1 & \alpha_3 & \cdots & \alpha_3 & \alpha_2 \end{pmatrix}$$

- **Good news:** we can compute the α_p in closed-form...
- ...and invert Σ, also in closed-form (lot of structure)

Inverting Σ

Proposition (G. and Mardaoui, 2021): Define $c_d := (d-1)\alpha_0\alpha_2 - d\alpha_1^2 + \alpha_0\alpha_1$, $\sigma_0 = (d-1)\alpha_2 + \alpha_1, \sigma_1 = -\alpha_1$,

$$\sigma_2 = \frac{(d-2)\alpha_0\alpha_2 - (d-1)\alpha_1^2 + \alpha_0\alpha_1}{\alpha_1 - \alpha_2}, \quad \text{and} \quad \sigma_3 = \frac{\alpha_1^2 - \alpha_0\alpha_2}{\alpha_1 - \alpha_2} .$$

Then the previous quantities are well-defined, $c_d > 0$, and Σ is invertible, with

$$\Sigma^{-1} = \frac{1}{c_d} \begin{pmatrix} \sigma_0 & \sigma_1 & \sigma_1 & \cdots & \sigma_1 \\ \sigma_1 & \sigma_2 & \sigma_3 & \cdots & \sigma_3 \\ \sigma_1 & \sigma_3 & \sigma_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \sigma_3 \\ \sigma_1 & \sigma_3 & \cdots & \sigma_3 & \sigma_2 \end{pmatrix} .$$

- *Proof:* algebra + four-letter identity. $\square$

Consequences

- **First consequence:** up to noise from the sampling, the explanations are linear in the model:

$$\beta^{f+g} \approx \beta^f + \beta^g \,.$$

- good property: we can split the explanations for additive models (linear models, random forests, kernel-based,...)
- **Second consequence:** simple expression in the *large bandwidth limit* ($\nu \to +\infty$):

$$\beta_j \approx 2 \left(\mathbb{E}\left[f(x) \mid z_j = 1 \right] - \mathbb{E}\left[f(x) \right] \right) \,.$$

- **Intuition:** large value if the model takes significantly larger values when superpixel j is present in the image
- this corresponds to our intuition

Shape detectors

- we can be *more precise* for specific models, for instance shape detectors:

$$f(x) = \prod_{u \in \mathcal{S}} \mathbb{1}_{x_u > \tau} \, ,$$

where $\mathcal{S}$ is a given set of pixels and τ is a positive threshold
- f takes value 1 if the shape $\mathcal{S}$ is lit up in image x
- we define the set of superpixels intersecting $\mathcal{S}$ as

$$E := \{ j \in \{1, \ldots, d\} \text{ s.t. } J_j \cap \mathcal{S} \neq \emptyset \} \, ,$$

which we split in two parts:

$$E_+ := \{ j \in E \text{ s.t. } \overline{\xi}_j > \tau \}, \text{ and } E_- := \{ j \in E \text{ s.t. } \overline{\xi}_j \leq \tau \}.$$

- for a given ξ, we also define

$$S_+ := \{ u \in \mathcal{S} \text{ s.t. } \xi_u > \tau \}, \text{ and } S_- := \{ u \in \mathcal{S} \text{ s.t. } \xi_u \leq \tau \}.$$

Shape detectors, ctd.

- with these notations in hand, we can compute β:

Proposition (G. and Mardaoui, 2021): Assume that $\forall j \in E_+, J_j \cap S_- = \emptyset$ and let $p := |E_-|$. Assume that, for all $j \in E_-$, $J_j \cap S_- = \emptyset$. Then, for any $j \in E_-$,

$$\beta_j^f = c_d^{-1} \left\{ \sigma_1 \alpha_p + \sigma_2 \alpha_p + (p-1)\sigma_3 \alpha_p + (d-p)\sigma_3 \alpha_{p+1} \right\}$$

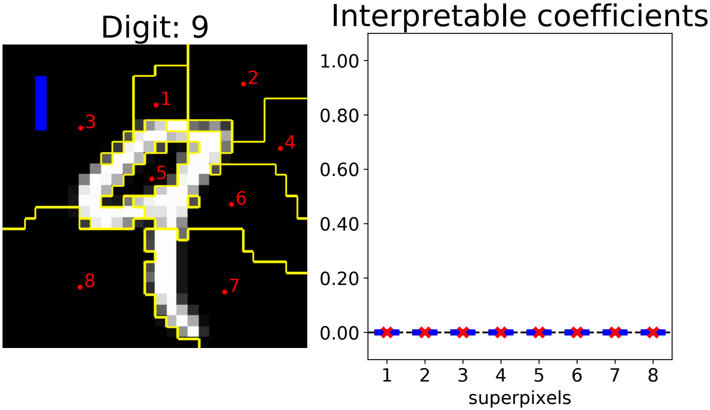
and for any $j \in \{1, \ldots, d\} \setminus E_-$,

$$\beta_j^f = c_d^{-1} \left\{ \sigma_1 \alpha_p + \sigma_2 \alpha_{p+1} + p\sigma_3 \alpha_p + (d-p-1)\sigma_3 \alpha_{p+1} \right\}$$

- simplifications when ν is large: $\beta_j \approx 1/2^{p-1}$ for an intersecting superpixel, 0 otherwise
- **Intuition**: LIME puts equal positive weights for superpixels intersecting $\mathcal{S}$

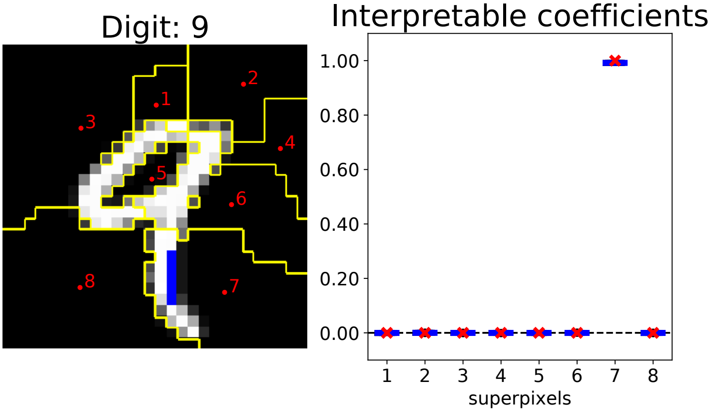
Shape detection example

- **Example:** rectangular shape, MNIST dataset, zero replacement:



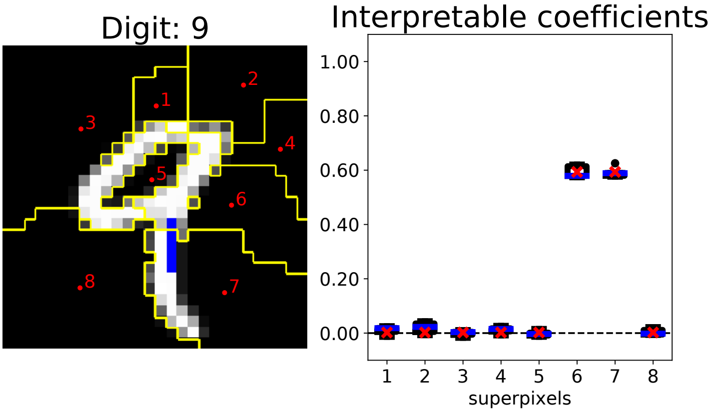
Shape detection example, ctd.

- **Example:** same digit, $\mathcal{S}$ intersects one superpixel:



Shape detection example, ctd.

- **Example:** same digit, $\mathcal{S}$ intersects two superpixels:



- **Take-away:** splitting the explanation between intersected superpixels

Linear models

- **Question:** what about linear models?
- let us set

$$f(x) = \sum_{u=1}^{D} \lambda_u x_u + b\,.$$

Proposition (G. and Mardaoui, 2021): assume that f is linear. Then

$$\forall 1 \leq j \leq d, \quad \beta_j = \sum_{u \in J_j} \lambda_u \cdot (\xi_u - \overline{\xi}_u)\,,$$

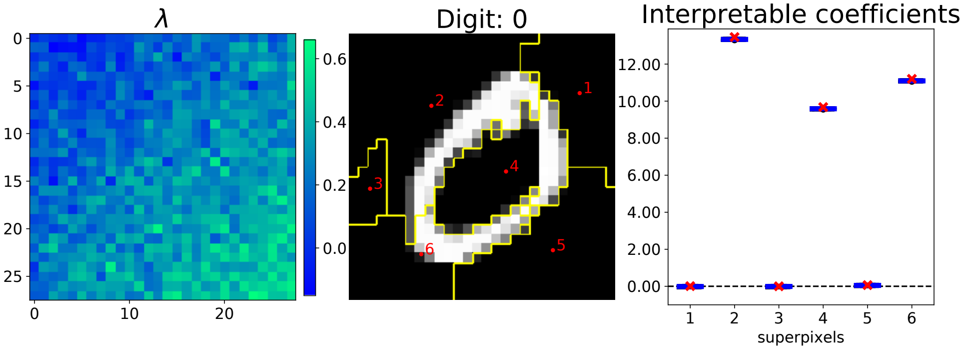
where $\overline{\xi}$ is the replacement image.

- **Intuition:** sum of gradient $\times$ input on the superpixels[9]

[9] Ancona et al., *Towards better understanding ¨ of gradient-based attribution methods for deep neural networks*, ICLR, 2018

Linear models, ctd.

- **Example:** linear function on MNIST with arbitrary coefficients:

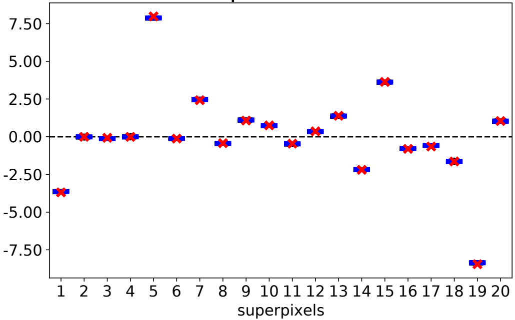


Linear models, ctd.

- **Example:** linear function on ILSVRC with arbitrary coefficients:

segmentation

first 20 interpretable coefficients



More complex models

- unsurprisingly difficult to extend the analysis
- **However,** if we replace f by a linear approximation, we see empirically that

$$\beta_j \approx \sum_{u \in J_j} \mathrm{IG}_u \cdot (\xi_u - \overline{\xi}_u) \,,$$

  where IG is the integrated gradients[10]
- $\approx$ averaged gradients of the model on a line joining ξ and a reference image

[10] Sundararajan, Taly, Qan, *Axiomatic attribution for deep networks*, ICML 2017

Integrated gradients

- **Idea:** average the gradient on a path between $\bar{\xi}$ and ξ and define[11]

$$\forall u \in \{1, \ldots, D\}, \quad \mathrm{IG}_u := \int_0^1 \frac{\partial f((1-\alpha)\xi + \alpha\bar{\xi})}{\partial x_u} \mathrm{d}\alpha$$

- approximate with Riemann sum:

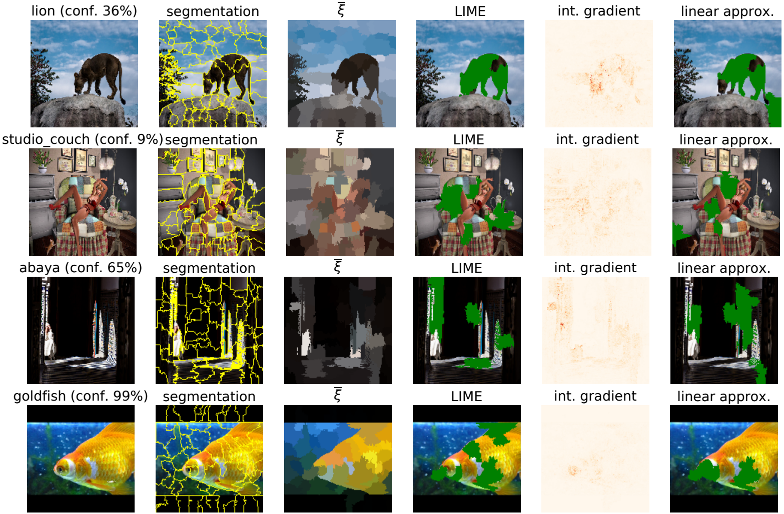
$$\mathrm{IG}_u^{\mathsf{approx}} := \frac{1}{m} \sum_{k=1}^{m} \frac{\partial f((1-\frac{k}{m})\xi + \frac{k}{m}\bar{\xi})}{\partial x_u} .$$

- linear approximation of f given by

$$f(x) \approx f(\bar{\xi}) + (x - \bar{\xi})^\top \mathrm{IG}^{\mathsf{approx}} .$$

[11] Sundararajan et al., *Axiomatic attribution for deep networks*, ICML, 2017

More qualitative results

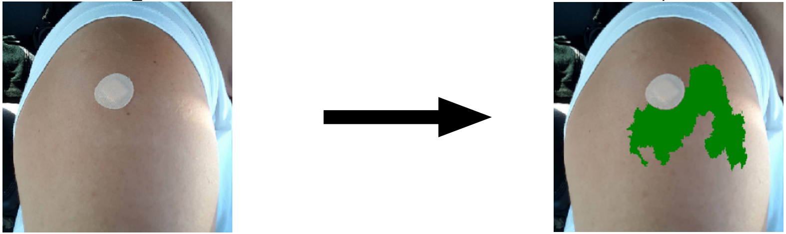


More qualitative results, ctd.

4. Conclusion

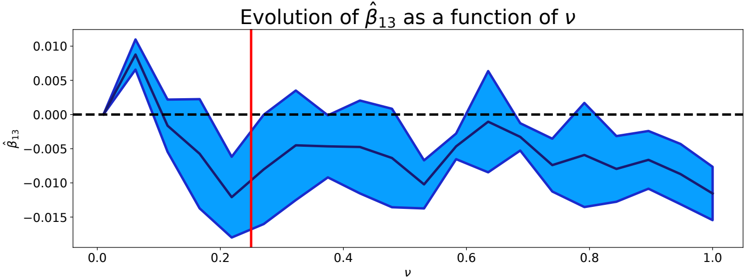
Some problems with LIME

- **Problem 1: the sampling**
- if the superpixel is very similar to the replacement superpixel, switching on and off does not change much
- LIME cannot learn that this pixel is important for the prediction
- even though it may be!



Some problems with LIME, ctd.

- **Problem 2: the bandwidth**
- ν is essentially the only free parameter of the method
- **Question:** what happens when we vary it?



- **Figure:** explanation for superpixel 13, ILSVRC dataset, InceptionV3 model, 10 repetitions for each ν, default is 0.25 (in red)
- **Undesirable behavior:** explanation changes sign when ν varies

Conclusion

- **In this talk:**
  - analysis of LIME for images
  - uncovering good properties (linearity, large bandwidth behavior)
  - but also less desirable ones, even for simple models proceed with caution!
- **Not in this talk:**
  - analysis for text[12] and tabular data[13,14]
  - similar message
- **Future directions:**
  - other methods, *e.g.*, Anchors[15] ($\approx$ rule extraction with similar sampling scheme)
  - general results for random local perturbation

[12] Mardaoui and Garreau, *An analysis of LIME for text data*, AISTATS, 2021
[13] Garreau and von Luxburg, *Explaining the explainer, a first theoretical analysis of LIME*, AISTATS, 2020
[14] Garreau and von Luxburg, *Looking deeper into tabular LIME*, arxiv, 2020
[15] Ribeiro et al., *Anchors: high-precision model-agnostic explanations*, AAAI, 2018

Thank you for your attention!